Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

Defining Acquisition Measures

The Integrated Software Acquisition Measurement Project (ISAM)

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Dr. John Mishler, NAVAIR Resident Affiliate, SEI, 2002-2003
Mr. Frank Sisti, Senior Member of the Technical Staff, SEI

Sponsored by the U.S. Department of Defense
© 2002 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Software Engineering Institute

DoD R&D Laboratory FFRDC

Sponsored by OUSD (AT&L)

Created in 1984

Under contract to Carnegie Mellon University

Offices in Arlington VA and Pittsburgh PA

Mission: Improve the practice of software engineering

© 2002 by Carnegie Mellon University

© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

JAC Challenge Problem for the SEI

The SEI Joint Advisory Committee (JAC)

- Is a tri-service oversight board to guide the SEI.
- Establishes SEI goals and direction.

One key challenge the JAC gave to the SEI is to define acquisition measurements to

- measure and manage software-intensive systems
- promptly, accurately, and precisely describe project status and trends
- support DoD program managers

© 2002 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

SEI's Acquisition Support Program

The SEI has established the Acquisition Support Program (ASP) to address system acquisition issues.

The Integrated Software Acquisition Measurement (ISAM) project is SEI's first step in addressing the acquisition measurement challenge.

The full Team Acquisition Process (TAP) effort will be a follow-on ASP project to address broader acquisition management needs.

© 2002 by Carnegie Mellon University

© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Acquisition Measurement Objectives

The ISAM project aims to develop integrated measures that

- apply to all development and acquisition levels
- provide broad life-cycle coverage
- promptly and precisely portray program status
- accurately predict future program performance
- minimally intrude on the development work
- support cyclic development
- facilitate process improvement
- are a natural consequence of quality work

The goal is to build a measurement culture at all levels of development and acquisition organizations.

© 2002 by Carnegie Mellon University Version 1.0 page 5

Carnegie Mellon
Software Engineering Institute

Metrics Program Requirements

To obtain useful measures, work must be precisely planned – without precise plans, work cannot be precisely tracked.

The development process must also be defined – undefined processes cannot be measured.

Process and product quality must be measured and managed – poor quality work makes projects late and unpredictable.

A useful metrics program must have people who consistently gather accurate data and know how to use these data.

© 2002 by Carnegie Mellon University Version 1.0 page 6

© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

The SEI Team Software Process

Software-intensive programs will not improve until the behavior of the software professionals changes.

The SEI has developed and is now transitioning the Team Software Process (TSP)SM into general practice.

With the TSP, precise measures are a basic and normal part of engineering practice.

The TSP provides the management and engineering training needed for rapid deployment and effective use of measures.

TSP projects predictably deliver the safe, secure, and high-quality software-intensive systems needed for modern warfare.

SM
Team Software Process and TSP are service marks of Carnegie Mellon University.

© 2002 by Carnegie Mellon University Version 1.0 page 7

Carnegie Mellon
Software Engineering Institute

The TSP Is Widely Used

Some of the organizations that are introducing and using the TSP are

- ABB
- AIS
- Bechtel
- Boeing
- Comnet
- DFAS
- EDS
- Ericsson
- Honeywell
- Iomega
- Kaiser
- Litton
- Microsoft
- NASA Langley
- SAIC
- SDRC
- Teradyne
- USAF: Hill AFB
- USN: NAVAIR
- USN: NAVOCEANO
- USN: NUWC
- Xerox

© 2002 by Carnegie Mellon University Version 1.0 page 8

© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

TSP Measurements

With the TSP, developers measure all of their work.
- time spent by phase
- size of products produced
- defects found by phase and product element

From these data, all required engineering project management measures can be derived.

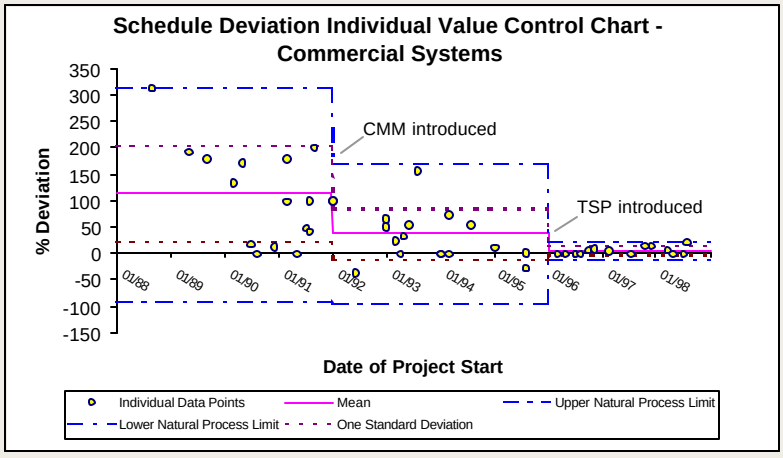
When using the TSP, development teams know precisely where their projects stand.

TSP teams regularly report on plan versus actual quality and schedule status, estimated project completion, and status of significant risks.

© 2002 by Carnegie Mellon University Version 1.0 page 9

Carnegie Mellon
Software Engineering Institute

Predictable Schedules



[Source: AIS]

© 2002 by Carnegie Mellon University Version 1.0 page 10

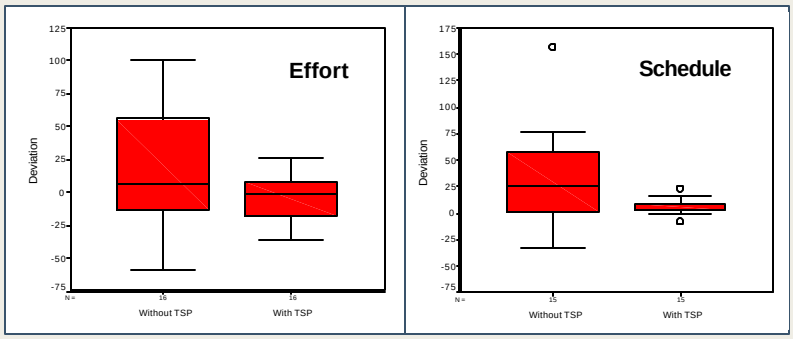
© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Cost and Schedule Performance

Sound budgets and commitments require accurate size and resource estimates.

Data from 24 teams in 4 organizations show that TSP teams make accurate estimates.



© 2002 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

TSP Quality Management

With the TSP, teams consistently improve product quality.

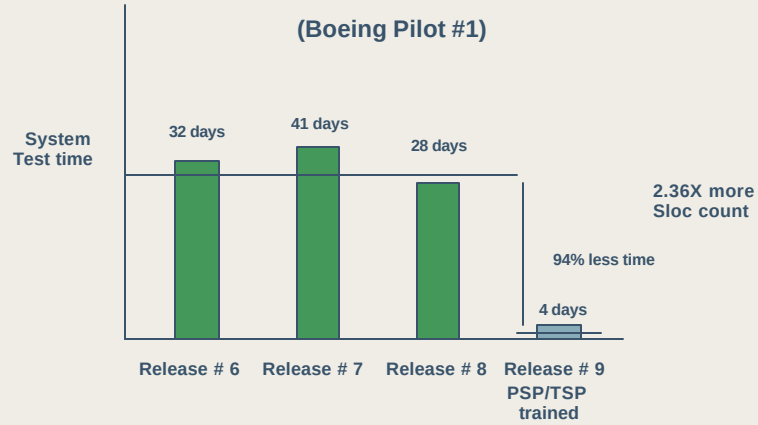
By doing quality work, TSP teams
- accelerate development schedules
- reduce program costs
- sharply cut testing time
- greatly reduce maintenance effort

© 2002 by Carnegie Mellon University

© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

TSP Quality Benefits - Boeing

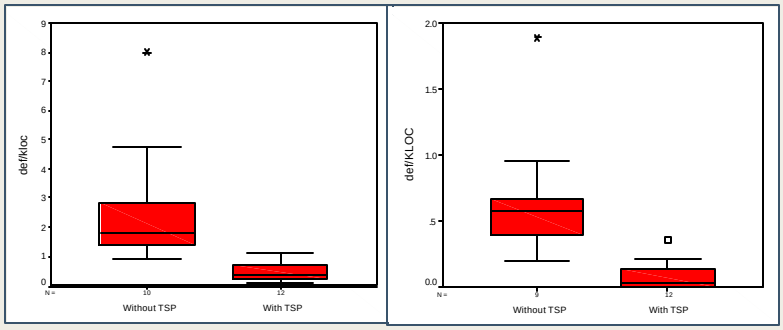


© 2002 by Carnegie Mellon University Version 1.0 page 13

Carnegie Mellon
Software Engineering Institute

Higher Product Quality

By planning, measuring, and tracking quality, TSP projects have fewer defects and shorter testing times.



System test defect density

Acceptance test defect density

© 2002 by Carnegie Mellon University Version 1.0 page 14

© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Reduced Cycle Time

Typical team

Reqts	Design	Implement	Test

TSP team

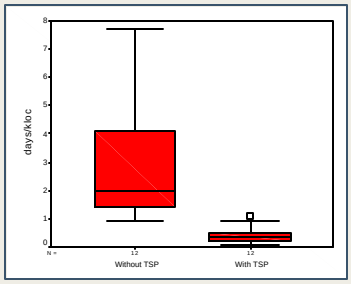
Reqts	Design	Implement	Test

Savings

With the TSP, teams find and fix defects early in the development process.

This sharply reduces test time.

With shorter testing, cycle time is cut.



Test Time per KLOC

© 2002 by Carnegie Mellon University Version 1.0 page 15

Carnegie Mellon
Software Engineering Institute

Reduced Development Costs

100 engineers
40 hours x 50 weeks
200,000 hours

Without TSP

40% test time =
80,000 hours of test

120,000 for development
2 LOC/hour = 240 KLOC

With TSP

10% test time =
20,000 hours of test

180,000 for development
2 LOC/hour = 360 KLOC

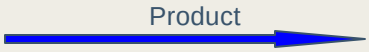
TSP = 50% Productivity Improvement

© 2002 by Carnegie Mellon University Version 1.0 page 16

© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

The ISAM Measurement Focus



Product Focus
- Customer Satisfaction (CUPRIMDSPS)
- Operational Capability
- Initial Operational Capability (IOC) Support
- Integrated Logistics Support (ILS) Plan
- System Integration (System of Systems)

Project Focus – ISAM
- Schedule (projection, status, trend)
- Cost (projection, status, trend)
- Defects (projection, status, trend)
- Project Cycle Time
- Total Ownership Cost (TOC)
- Cost of Quality
- Requirements Satisfaction

Other Focus
- Legal
- Financial
- Contracts
- Customer
- Sponsor

Process Focus
- Cycle time trends
- Support Capability
- Total Ownership Cost trends
- Requirements Management

© 2002 by Carnegie Mellon University Version 1.0 page 17

Carnegie Mellon
Software Engineering Institute

Effective Program Measurement

To be useful for management, measures must be
- precisely defined, accurate, and traceable
- timely and predictive
- used by engineering
- a minimum complete and consistent set

The TSP accomplishes this by providing comprehensive measurements that are integral to the engineering work.

With TSP, measurement is natural. The TSP
- precisely and promptly measures the engineering work
- does not impose added measurement costs
- provides a defined and non-proprietary measurement system

© 2002 by Carnegie Mellon University Version 1.0 page 18

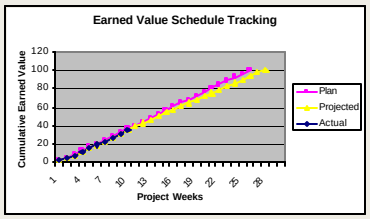
© 2001 by Carnegie Mellon University

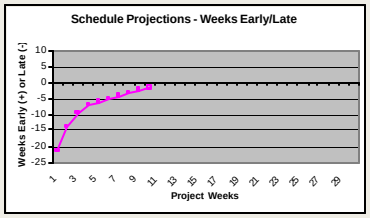
Carnegie Mellon
Software Engineering Institute

Schedule Predictability

The schedule predictability measures are

- earned value
- completion projections
- completion projection trends





© 2002 by Carnegie Mellon University Version 1.0 page 19

Carnegie Mellon
Software Engineering Institute

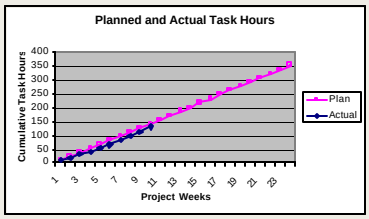
Effort Predictability

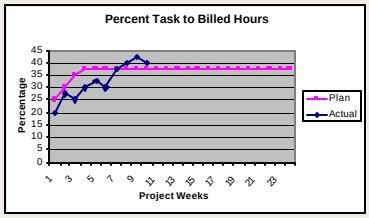
Actual versus planned task hours

Billed hours versus task hours

Relationship of task hours and EV status

Effort estimate accuracy for completed tasks





© 2002 by Carnegie Mellon University Version 1.0 page 20

© 2001 by Carnegie Mellon University

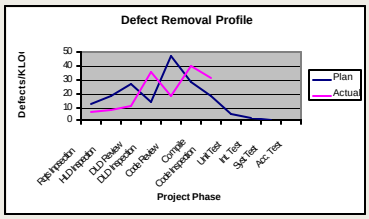
Carnegie Mellon
Software Engineering Institute

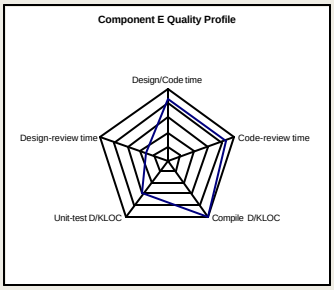
Quality

Defects per KLOC, planned and actual

Percent defect free (PDF)

Quality profiles





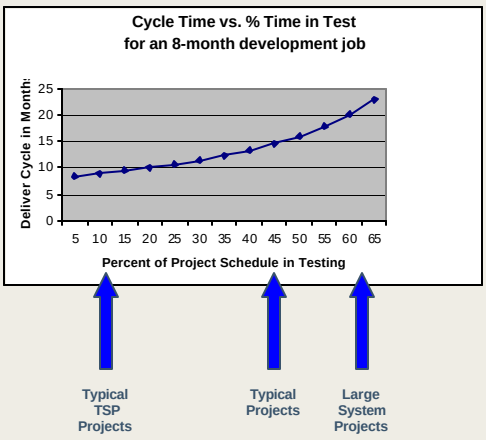
Carnegie Mellon
Software Engineering Institute

Cycle Time Reduction

Time in days from project initiation to initial operational capability

Time to first production article delivery

Percent of cycle time spent in test phases (after unit testing)



© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Other Measures

Total Ownership Cost
- Cost of program development (program initiation to initial operational capability)
- Cost of program maintenance (cost to maintain product after initial operational capability

Cost of Quality
- Percent of total development time spent in appraisal (walkthroughs, reviews, and inspections)
- Percent of total development time spent in rework (compile and test)

Requirements Satisfaction
- Number of acceptance test defects in user acceptance or operational suitability tests
- Acceptance test defects per KLOC

© 2002 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Required Training

To adopt these measures, acquisition groups will require a few days of ISAM training.

The development groups must use the TSP.

An extensive program is available for transitioning the TSP into development and maintenance organizations.
- Executive kickoff and planning seminar – 1 1/2 days
- Management training – 2 days
- Engineer training: 2 week PSP course
- Internal transition agent training
  - PSP Instructor – 5 days
  - TSP Launch Coach – 5 days

Training costs are recovered with the first 1,000 LOC developed.

© 2002 by Carnegie Mellon University

© 2001 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Next Steps

Complete program management interviews.

Refine proposed measures.

Establish metrics-based management methods.

Define metrics prototype testing effort.

Conduct prototype tests
- Program managers use ISAM measurements.
- Projects use the TSP.

Produce final report and transition plan.

© 2002 by Carnegie Mellon University Version 1.0 page 25

Carnegie Mellon
Software Engineering Institute

Conclusions

Improved measurements are needed to manage software-intensive systems throughout their life cycle.

The TSP provides the foundation for precise and timely program measurements.

ISAM provides the measurement tools for effective and responsive program management.

With your help and support, this project will guide future program managers in meeting our military's needs for reliable, safe, and secure software-intensive systems.

© 2002 by Carnegie Mellon University Version 1.0 page 26

© 2001 by Carnegie Mellon University